REMARKS

Claims 1-30 are currently pending. Applicant respectfully requests reconsideration of this application. Claims 2, 3, 5-8, 12, 13, 16-21, 23-25, and 27 have been amended. No claims have been cancelled or added.

Therefore, claims 1-30 are now presented for examination.

Amendments to the Specification

An amendment to paragraph 0027 has been made to correct a typographical error.

Claim Rejection under 35 U.S.C. §103

Klimenko, et al. in view of Haskins, et al.

The Examiner rejected claims 1-30 under 35 U.S.C. 103 (a) as being unpatentable over U.S. Patent 5,974,547 of Klimenko, et al. (hereinafter referred to as "*Klimenko*") in view of U.S. Patent 6,240,169 of Haskins, et al. (hereinafter referred to as "*Haskins*").

In response, claim 1 has been clarified to assist in addressing this issue. Claim 1, as amended herein, is as follows:

1. A method comprising:
   requesting a memory address region and network boot load data from a server;
   receiving the network boot load data and a designated memory region from the server; and
   loading the network boot load data into the designated memory region.

Klimenko involves a form of network booting of an operating system to a client computer. In particular, *Klimenko* provides for storing an image of a client hard drive, including operating system. This image is then accessed by a hard disk emulation process. As described in the Summary of *Klimenko*, there is a continuous client hard disk emulation that exists throughout the boot process. (*Klimenko*, col. 3, lines 50-57)

However, what is described in *Klimenko* is a different kind of process than is described in claim 1. While *Klimenko* refers to "network booting" of an operating system, it is referring to the location of the operating system and applications, not to the boot load data. *Klimenko* utilizes a network to present an image of client's hard drive, and then uses this image to load the operating system. *Klimenko* does not address the obtaining boot data – the data required to boot a system – from the network, but is rather involved in the question of whether the operating system and applications are stored.

Klimenko thus is actually addressing a different kind of issue than Claim 1, which relates which is the storing of network boot load data received from the server (which then may be used to initiate the operating system). *Klimenko* does not describe a system in which a memory address region and network boot load data are requested and received from a server, and in which the network boot load data is loaded into the designated memory region. *Klimenko* instead describes a system in which a bootloader is conventionally downloaded and used for the boot process.

The Examiner cites to certain portions of *Klimenko* with regard to the claim elements, including the following (with additional text provided to provide context) with regard to Figure 1:

> In this context, FIG. 1 depicts a high-level simplified block diagram of client-server environment 5 in which client PC 10 is to be

> booted through server 50. As shown, client PC 10 is connected, via links 20 and 40, and network 30, to network server 50. Inasmuch as the particular implementation and architecture of network 30 are both irrelevant, the ensuing discussion will omit all such details. Through the present invention, a complete image of the Windows 95 O/S that is to execute on client PC 10 is stored as image 56 on hard disk 54 within memory 52 of server 50.
>
> In operation, whenever a user energizes (powers-up) client PC 10, this PC then establishes a network connection to the server and issues a boot request, as symbolized by line 62, to the server. In response to this request, as symbolized by line 64, the server downloads sufficient files from the stored client O/S image to the client PC to permit the client to boot the O/S and continue loading the required O/S files from the server.

(*Klimenko*, col. 6, lines 19-36) As indicated, the operating system that is to execute on a client PC is stored as an image within the memory of the server. When a user powers up the client PC, the PC then establishes a network connection to the server, and issues a "boot request". As indicated, the server then "downloads sufficient files from the stored client O/S image to the client PC to permit the client to boot the O/S and continue loading the required O/S files from the server." However, there is no indication in this text that a memory address region or network boot data is requested from the server. Instead, server is delivering sufficient files to permit the client to boot the O/S (as stored on the network) and continue loading the required O/S.

In fact, it is submitted that the text of *Klimenko* would indicate that a more conventional source of boot data is used. For example, *Klimenko* provides the following in describing Figure 2A:

As shown, client PC 10 contains LAN adapter (also commonly referred to as a network interface card--NIC) 360. Each such NIC carries a unique physical hardware address, referred to as a media access control (MAC) address, through which that card can be uniquely addressed on a network. An illustrative MAC address is "00A024Baf9a5". Each NIC also contains internal read only memory 362 that stores boot code 364, which contains a BootP client process. Though this code is usually stored within the NIC, as shown here, this code could alternatively be implemented within a PC ROM BIOS (basic input output system) located on a motherboard of the client PC. With the boot code stored in the NIC, as shown, and read into memory of the PC on power-up and executed, the client PC establishes a network connection, through network 30 and connections 20 and 40, with remote server 50 for remotely booting of the client PC. Server 50 contains, to the extent relevant to the present invention, TCP (transmission control protocol) servers 230, specifically: either BootP server 232 or DHCP (dynamic host configuration protocol) server 234, and my inventive random access trivial file transfer protocol (RATFTP) server 236. The BootP and DHCP servers are conventional in nature and, as such, will not be discussed in any detail. On the other hand, the RATFTP server, while based on and extends capabilities of a conventional trivial file transfer protocol (TFTP) server, accesses individual desired sector(s) (rather than just a complete file as does a conventional TFTP server), on hard disk 54 situated within server 50--thus facilitating client hard disk emulation. Such sectors are specified by a boot loader and downloaded into client PC during the network boot process.

(*Klimenko*, col. 7, lines 11-41) (emphasis added) The process shown by *Klimenko* does not provide for any request for a memory address region, and does not provide for the loading boot load data into the requested memory address region. Instead, what is described is a system in which a boot code is accessed from the NIC (network interface

card) or from PC ROM BIOS (basic input output system). This boot code is then read into memory, and provides a network connection for remotely booting up the client PC. The boot code includes a BOOTP client process, which apparently is a reference the known Bootstrap Protocol (BOOTP) described in Internet RFCs 951 and 1084, which is used for booting diskless workstations, enabling the workstation to find its own logical IP address at startup. Thus, to the degree that *Klimenko* refers to boot data, it does not refer to receipt of such data from a network, but rather from a network interface card or ROM BIOS.

The other cited portions of *Klimenko* are also irrelevant to the current claims. For example, for the element of storing boot image data in a memory the Examiner has cited to a sentence in the following paragraph:

> As shown, client PC 10 comprises input interfaces (I/F) 310, processor 320, NIC 360, memory 330 and output interfaces 340, all conventionally interconnected by bus 350. Memory 330, which generally includes different modalities, includes illustratively random access memory (RAM) 332 for temporary data and instruction store, diskette drive(s) (not specifically shown) for exchanging information, as per user command, with floppy diskettes, and non-volatile mass store 335 that is implemented through hard disk drive(s) 334, typically magnetic in nature. Should client PC 10 be implemented by "diskless" computer, then all disk drives, including both floppy diskette drive(s) and hard disk drive(s) 334, would be omitted. Regardless of whether client PC 10 contained a hard disk drive or not, the client O/S, during its boot process, would be downloaded into RAM 332 and executed therefrom. As shown above in FIG. 2A, NIC 360 contains internal read-only memory 362, that stores network boot code 364. This code, as will be discussed shortly below,

> once downloaded into RAM 332 on power-up permits the NIC to establish a network connection to a remote server.

(*Klimenko*, col. 7, lines 11-41) (emphasis added) Thus, *Klimenko* is discussing downloading the client operating system, not network boot data. The "network boot code" is instead stored in internal read-only memory in the network adaptor.

It is submitted that *Haskins* fails to teach or suggest the elements of claim 1 that are missing from *Klimenko*. *Haskins* deals with the different issue of a least call routing system, and specifically with regard to choosing which of a number of telephone routes will result in the lowest cost. The Examiner has cited this reference specifically with regard to a memory address region. *Haskins* fails to teach or suggest the elements of requesting a memory address region and network boot data from a server, receiving the network boot load data and a designated memory region from the server, or loading the network boot load data into the designated memory region. *Haskins* is focused on a very different technology, and does not appear to be relevant regarding any of these claim elements.

Klimenko and *Haskins*, alone or in combination, thus do not contain the elements of claim 1, as amended. It is submitted that the arguments presented herein also apply to independent claims 8, 12, 19, and 27, and thus such claims are allowable for similar reasons. The remaining rejected claims are dependent claims that, while having other independent reasons for allowance, are allowable as being dependent on the allowable base claims.

Because of the other arguments presented here have disposed of the issues and demonstrate that obviousness has not been shown, all additional arguments are not

provided. However, Applicant respectfully submits that *Klimenko* and *Haskins*, which involve disparate and largely unconnected technologies, have not been properly combined as references, and, regardless of the teachings of each reference, cannot be relied upon for this rejection.

Conclusion

Applicant respectfully submits that the rejections have been overcome by the amendment and remark, and that the claims as amended are now in condition for allowance. Accordingly, Applicant respectfully requests the rejections be withdrawn and the claims as amended be allowed.

Invitation for a Telephone Interview

The Examiner is requested to call the undersigned at (503) 439-8778 if there remains any issue with allowance of the case.

Request for an Extension of Time

The Applicant respectfully petitions for a one-month extension of time to respond to the outstanding Office Action pursuant to 37 C.F.R. § 1.136(a). A check is enclosed to cover the necessary fee under 37 C.F.R. § 1.17 for such an extension.

Charge our Deposit Account

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 10/23/06

Mark C. Van Ness
Reg. No. 39,865

12400 Wilshire Boulevard
7th Floor
Los Angeles, California 90025-1026
(303) 740-19800

The PTO did not receive the following listed item(s) a check